



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

Address: COMMISSIONER FOR PATENTS

P.O. Box 1450

Alexandria, Virginia 22313-1450

www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/730,192	12/04/2003	Curt L. Cotner	SVL920030122US1/2991P	2148
7590 01/15/2010 SAWYER LAW GROUP LLP P.O. Box 51418 Palo Alto, CA 94303				
EXAMINER MORRISON, JAY A				
ART UNIT		PAPER NUMBER		
2168				
MAIL DATE		DELIVERY MODE		
01/15/2010		PAPER		

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte CURT L. COTNER, TAMMIE DANG, BRIAN K. HOWELL,
FUNG LEE, HUI-AN LEE, CHARLES H. LIN, CLAIRE W. MCFEELY,
JENNIFER W. OU, JAMES W. PICKEL, and JAY A. YOTHERS

Appeal 2008-5056
Application 10/730,192
Technology Center 2100

Decided: January 15, 2010

Before LEE E. BARRETT, JOHN A. JEFFERY, and
DEBRA K. STEPHENS, *Administrative Patent Judges*.

JEFFERY, *Administrative Patent Judge*.

DECISION ON APPEAL

Appellants appeal under 35 U.S.C. § 134(a) from the Examiner's rejection of claims 1-18. We have jurisdiction under 35 U.S.C. § 6(b). We affirm.

STATEMENT OF THE CASE

Appellants' invention provides package resolution in a database system. Specifically, a statement is provided in a server that allows an application to identify a list of package collections, and the statement is executed to obtain the appropriate package. Moreover, a new current package path value associated with a list of package collections is set so that a list of qualified package collections is supplied during runtime.¹ Claim 1 is illustrative:

1. A method for providing package resolution in a database system, the method comprising:

providing a database statement within a server of the database system, the database statement allowing an application to identify a list of package collections, wherein each identified package collection includes a plurality of packages and each package is usable during execution of one or more other database statements issued by the application;

responsive to issuance of the database statement by the application, executing the database statement to locate at least one of the plurality of packages included in at least one of the identified package collections; and

caching the at least one package in a storage of the server, the at least one package being used during execution of the one or more other database statements issued by the application.

The Examiner relies on the following as evidence of unpatentability:

Bird

US 6,321,235 B1

Nov. 20, 2001

¹ See generally Abstract; Spec. 5:15-24.

THE REJECTION

The Examiner rejected claims 1-18 under 35 U.S.C. § 102(b) as anticipated by Bird. Ans. 3-6.²

Regarding representative claim 1,³ the Examiner finds that Bird discloses a method for providing package resolution with all of the recited subject matter. Ans. 3-4. According to the Examiner, Bird's SQL request corresponds to the recited database statement, and that this request is satisfied by a static SQL cache or is routed to a dynamic SQL cache. These caches are said to correspond to the recited list of package collections since the static cache comprises multiple packages which contain multiple sections. Ans. 7. The Examiner takes a similar position regarding the dynamic cache since it is said to be partitioned into anchor points that are further divided into partitions or portions (i.e., compilation environments). *Id.*

Appellants argue that while Bird mentions packages, it does not teach or suggest a database statement allowing an application to identify a list of package *collections*, where each identified package collection includes plural packages as claimed. Br. 5-6 (emphasis added).

² Throughout this opinion, we refer to the Appeal Brief filed December 19, 2007 and the Examiner's Answer mailed August 11, 2008.

³ Appellants argue two claim groupings separately: (1) claims 1, 6, and 11; and (2) claims 2, 7, and 12. *See* Br. 4-8. Since claims 3-5, 8, 9, and 13-18 were not separately argued, we group these claims with group (1), and select claim 1 as representative of that group. Also, since claim 10 depends from claim 7, we group claim 10 in group (2), and select claim 2 as representative of that group. *See* 37 C.F.R. § 41.37(c)(1)(vii).

Representative claim 2 calls for the database statement to be a “SET CURRENT PACKAGE PATH” statement. Appellants argue that the Examiner’s reliance on the “CURRENT FUNCTION PATH” in Bird is misplaced since it is not a database statement, but rather a “special register.” Br. 7-8. Appellants add that even if it could be construed as a database statement, it does not correspond to the recited statement. Br. 8.

The Examiner takes the position that the claim merely recites a “statement fragment” and does not recite its functionality. As such, the Examiner contends, the claim is fully met by Bird’s “statement fragment” which refers to a path. Ans. 9.

The issues before us, then, are as follows:

ISSUES

Under § 102, have Appellants shown that the Examiner erred by finding that Bird discloses:

(1) a database statement allowing an application to identify a list of package collections, where each identified package collection includes plural packages as recited in claim 1?

(2) a “SET CURRENT PACKAGE PATH” database statement as recited in claim 2?

FINDINGS OF FACT

The record supports the following findings of fact (FF) by a preponderance of the evidence:

Appellants' Disclosure

1. According to Appellants' Specification:

The application 12 includes host language and SQL statements which are utilized to communicate with the server to obtain the appropriate information from the tables therein. The database server 14 typically prepares or converts the host language to allow the application to be executed (by the server). This preparation or process is referred to as preparing a package. A package is typically identified uniquely by an application program identifier (application ID) and a collection identifier (collection ID).

Spec. 1:11-17.

2. Appellants' Specification indicates:

A system and method in accordance with the present invention provides a new current package path value, which is set via a statement referred to as SET CURRENT PACKAGE PATH. The SET CURRENT PACKAGE PATH statement allows the application to supply a precedence list of qualified package collections during execution time, therefore the scheme for resolving packages can be changed at any point during execution.

Spec. 4:12-17.

3. According to Appellants' Specification, the SET CURRENT PACKAGE PATH is preferably an SQL statement, but "[o]ne of ordinary skill in the art readily recognizes . . . that the statement could be in a variety of languages and its use would still be within the spirit and scope of the present invention." Spec. 5:15-20.

4. Appellants' Specification notes that the SET CURRENT PACKAGE PATH statement "allows the application to identify a precedence list of package collections that is used during package resolution.

The CURRENT PACKAGE PATH value contains a list of collections for packages. The value is maintained and updated through the SET CURRENT PACKAGE PATH statement.” Spec. 5:20-24.

Bird

5. Bird discloses a relational database management system (RDBMS) 10 involving caching dynamic and static Structured Query Language (SQL) statements so that these statements can be reused by multiple requesters. Bird, col. 1, ll. 6-16; col. 3, ll. 22-29; Fig. 1.

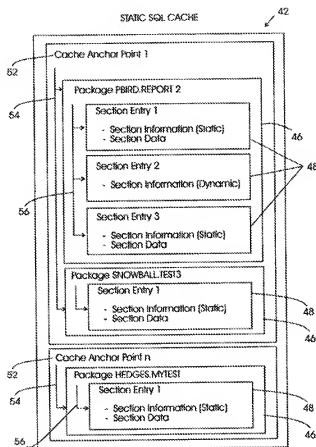
6. A node 16 of the database system may contain any number of applications 30 with respective SQL work areas 32, each application accessing data from the database(s) to which the node is connected. Bird, col. 3, ll. 45-51; Fig. 2.

7. Node 16 also comprises a global cache 40 which acts as a repository for (1) package and section information for static SQL statements, and (2) section information for dynamic SQL statements. Bird, col. 3, ll. 51-56; Fig. 2.

8. Global cache 40 consists of two logical areas: (1) the static SQL cache 42, and (2) the dynamic SQL cache 44. Bird, col. 5, ll. 12-14; Fig. 3.

9. The static cache 42 contains the information for packages 46, section entries 48, and static SQL sections 40. Within each package, a section entry 48 associated with a specific package has a specific assigned section number indicating its location within the package. Bird, col. 1, ll. 45-51; col. 5, ll. 14-25; Fig. 3.

10. Each package 46 can have multiple associated section entries 48.
Bird, col. 5, ll. 43-44; Fig. 4. Bird's static SQL cache is shown in Figure 4 reproduced below:



Bird's Static SQL Cache in Figure 4

11. To facilitate access, each package 46 is associated with a specific static cache anchor point 52. Then, the package is inserted into the list of packages 54 from the static anchor point 52 in alphabetical order based on package name and qualifier. Bird, col. 5, ll. 47-53; Fig. 4.

12. Internally, the static SQL cache 42 consists of a static SQL control block containing a list of cache anchor points 52. Each cache anchor

point 52 points to a series of two-dimensional linked lists of package/section information (i.e., (1) a linked list of pointers to the packages, and (2) a linked list of section entries within each package). Bird, col. 5, ll. 58-64.

13. Each linked list of packages 54 hangs from an anchor point 52, and the base for all anchor points is in the control block. Bird, col. 5, ll. 64-66.

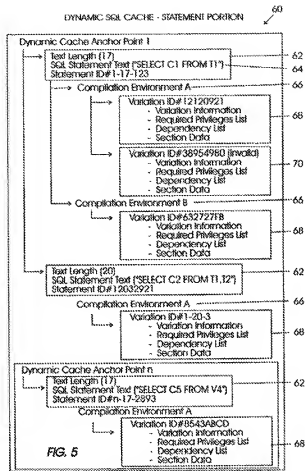
14. According to Bird:

A package is associated with an application and contains the information required by the RDBMS for all SQL statements defined in that application. The information in a package consists of a collection of sections and the compilation environment settings (e.g., compile or binding options) used to compile any static SQL statements; some of these same settings are also used as the default environment for any dynamic SQL statements compiled by the application during execution.

Bird, col. 1, ll. 35-44.

15. The dynamic SQL cache 44 comprises SQL statement entries 62 and associated compilation environments 66 that contain information on the environment used to generate or compile the section. Below each compilation environment 66 are stored individual “variation” units representing unique sections for the statement text specified by the compilation environment. Bird, col. 6, l. 8 – col. 7, l. 5; Figs. 3 and 5.

16. Figure 5 shows that the statement portion of the dynamic cache comprises a number of anchor points associated with respective SQL statement entries and their corresponding compilation environments and variations. Bird’s Figure 5 is reproduced below.



Bird's Statement Portion of the Dynamic Cache in Figure 5

17. Bird provides an example of a variation 68 in the context of a particular dynamic SQL statement, "SELECT C1 FROM T1." Bird notes that if an SQL statement contains an unqualified function reference (e.g., SELECT FOO(C1) FROM T1), then anyone issuing the exact same statement with the exact same value in the CURRENT FUNCTION PATH special register will use the exact same function. Bird, col. 7, ll. 16-29, 44-60.

PRINCIPLES OF LAW

Anticipation is established only when a single prior art reference discloses, expressly or under the principles of inherency, each and every element of a claimed invention as well as disclosing structure which is capable of performing the recited functional limitations. *RCA Corp. v. Appl. Dig. Data Sys., Inc.*, 730 F.2d 1440, 1444 (Fed. Cir. 1984); *W.L. Gore & Assoc., Inc. v. Garlock, Inc.*, 721 F.2d 1540, 1554 (Fed. Cir. 1983).

During patent examination, a claim is given its broadest reasonable construction “in light of the specification as it would be interpreted by one of ordinary skill in the art.” *In re Am. Acad. of Sci. Tech. Ctr.*, 367 F.3d 1359, 1364 (Fed. Cir. 2004). Such an interpretation, however, must not import limitations from the Specification into the claims. “[A]lthough the specification often describes very specific embodiments of the invention, we have repeatedly warned against confining the claims to those embodiments. . . [C]laims may embrace different subject matter than is illustrated in the specific embodiments in the specification.” *Phillips v. AWH Corp.*, 415 F.3d 1303, 1323 (Fed. Cir. 2005) (en banc) (citations and internal quotation marks omitted).

ANALYSIS

We begin by construing the key limitation of claim 1 which calls for, in pertinent part, a database statement that allows an application to identify a list of *package collections*, where each identified package collection includes plural packages. To this end, we first construe the term “package.”

Appellants' Specification does not explicitly define the term "package" as claimed, but does note that preparing or converting an application's host language to enable its execution is referred to as "preparing a package." FF 1.

Bird, however, defines the term "package" as "associated with an application and contains the information required by the RDBMS for all SQL statements defined in that application." FF 14. Bird further notes that "[t]he information in a package consists of a collection of sections and the compilation environment settings (e.g., compile or binding options) used to compile any static SQL statements." *Id.*

Although Bird's definition of "package" does not limit our construction, it nevertheless informs our understanding of the term, particularly in light of the lack of a clear definition in Appellants' Specification. Based on the record before us, we construe the term "package" as a module or logic associated with an application that contains information associated with executing the application.

Based on this construction, we see no error in the Examiner's position (Ans. 7-8) that Bird's static SQL cache fully meets the recited list of "package collections." As the Examiner indicates (*id.*), Bird's static SQL cache comprises multiple packages that are further divided into section entries. FF 9-10. Notably, nothing in the claim precludes the packages' section entries from constituting a "package" given the scope and breadth of the term. In essence, the section entries are effectively "sub-packages" in view of their hierarchical arrangement with respect to their associated packages as shown in Bird's Figure 4. See FF 9-10.

In that sense, we see no error in the Examiner's position that Bird's static SQL cache essentially constitutes a list of package collections (i.e., a list of packages and their associated "sub-packages"). We reach a similar conclusion regarding the Examiner's position (Ans. 7-8) pertaining to Bird's dynamic cache which also uses a hierarchical arrangement of statement entries and their corresponding compilation environments and variations (i.e., "packages" and "sub-packages"). *See* FF 15-16.

Furthermore, as shown in Bird's Figure 4, each package in the static SQL cache is associated with a specific static cache anchor point that delineates an associated list of packages. FF 11. Notably, each linked list of packages is associated with a particular anchor point (FF 13), and the static SQL cache's control block contains a list of cache anchor points (FF 14). We see no reason why this list of anchor points associated with the static SQL cache would not constitute a list of package collections as claimed, particularly since each anchor point provides a list of associated packages (i.e., each anchor point has an associated package collection). *See* FF 10-13.

For the foregoing reasons, Appellants have not persuaded us of error in the Examiner's rejection of claim 1, and claims 3-6, 8, 9, 11, and 13-18 which fall with claim 1.

Claims 2, 7, 10, and 12

We will also sustain the Examiner's rejection of representative claim 2 which calls for the database statement to be a "SET CURRENT PACKAGE PATH" statement. We note at the outset that the claim merely recites the name of a statement, but does not recite its corresponding functionality. In general, such non-functional recitations constitute non-

functional descriptive material as they do not further limit the claimed invention either functionally or structurally. Such non-functional descriptive material does not patentably distinguish over prior art that otherwise renders the claims unpatentable.⁴

That said, we acknowledge that the “SET CURRENT PACKAGE PATH” statement is intended to set a “CURRENT PACKAGE PATH” value associated with a list of package collections so that a list of qualified package collections is supplied during runtime. FF 2 and 4. But claim 2 does not recite this disclosed functionality, nor will we import this functionality into the claim. *See Phillips*, 415 F.3d at 1323.

With this construction, we turn to Bird. Although Appellants are correct (Br. 8) that Bird’s “CURRENT FUNCTION PATH” pertains to a register (FF 17), there is nonetheless a database statement in Bird (or “statement fragment” to use the Examiner’s term (Ans. 9)) associated with this register. And this statement would, at least in part, pertain to a path in connection with the associated “package” of the dynamic SQL cache. *See* FF 15-17. Although this statement may not be labelled as a “SET CURRENT PACKAGE PATH” statement, it nonetheless fully meets claim 2 given its scope and breadth. That Appellants’ Specification emphasizes that the “SET CURRENT PACKAGE PATH” statement “could be in a variety of languages” (FF 3) only bolsters our conclusion that the form of the statement itself is not critical.

⁴ *See In re Ngai*, 367 F.3d 1336, 1339 (Fed. Cir. 2004); *see also Ex parte Nehls*, 88 USPQ2d 1883, 1887-89 (precedential) (discussing cases pertaining to non-functional descriptive material).

For the foregoing reasons, Appellants have not persuaded us of error in the Examiner's rejection of claim 2, and claims 7, 10, and 12 which fall with claim 2.

CONCLUSION

Appellants have not shown that the Examiner erred in rejecting claims 1-18 under § 102.

ORDER

The Examiner's decision rejecting claims 1-18 is affirmed.

No time period for taking any subsequent action in connection with this appeal may be extended under 37 C.F.R. § 1.136(a)(1)(iv).

AFFIRMED

pgc

SAWYER LAW GROUP LLP
P.O. Box 51418
Palo Alto CA 94303